

A Bus Arbitration Scheme with an Efficient Utilization and Distribution

Amin M. A. El-Kustaban

Department of Electronic Engineering
University of Science and Technology (UST)
Sana'a, Yemen

Abdullah A. K. Qahtan

Department of Electronic Engineering
University of Science and Technology (UST)
Sana'a, Yemen

Abstract—Computer designers utilize the recent huge advances in Very Large Scale Integration (VLSI) to place several processors on the same chip die to get Chip Multiprocessor (CMP). The shared bus is the most common media used to connect these processors with each other and with the shared resources. Distributing the shared bus among the contention processors represents a critical issue that affects overall performance of the CMP. Optimal utilization with fair distribution of the shared bus represents another challenge. This paper introduces a bus arbitration scheme, which is an Age-Based Lottery (ABL) Arbitration that combines the lottery and age-based algorithms to overcome the shared bus challenges. The results show that the developed bus arbitration scheme maximizes the bus utilization and improves the distribution by at least 13.5% with an acceptable latency time comparing to the traditional bus arbitration schemes.

Keywords—Chip Multiprocessor; Round Robin; Lottery Algorithms; Latency; VHDL

I. INTRODUCTION

The technology revolution in Very Large Scale Integration (VLSI) has enabled today's designers to design and implement Chip Multiprocessor (CMP), where two or more processors with a shared memory are integrated on a single chip [1].

The contention between the processors in CMP systems adds significant overhead in order to manage the access to that shared bus [2]. Thus, scheduling mechanisms or "arbitration schemes", which are employed to synchronize and schedule the bus requesting from different bus masters in order to avoid contentions, have a major and important effect on the overall performance of the CMP design [3-5]. One of the challenges faced by the bus arbitration is to ensure that the sharing resources can be utilized and balanced distribution among the contention masters.

The improvements on the bus arbitration protocols are performed to enhance some of the protocols' aspects, such as: the fairness degree, latency time, bandwidth utilization, responding to priorities, cost, and power consumption [6].

In this paper, a bus arbitration scheme, which is called an Age-Based Lottery (ABL), is introduced. This scheme overcomes the static and dynamic lottery schemes shortcomings such as the unbalance distribution of the bus. Also, this paper improves the performance by maximizing the shared bus utilization and balancing the bus distribution with an acceptable latency. The results are shown and compared to

the traditional bus arbitration schemes by implementing them using the Hardware Description Language (HDL) and illustrating the testing results using ModelSim tool.

This paper is organized as follows. Section II, reviews the related work. Section III, introduces the most knowing bus arbitration schemes. Section IV, discusses the developed bus arbitration scheme. Implementing, testing and comparing the developed bus arbitration scheme to the traditional schemes are presented in section V. Finally, conclusion and future work are summarized in section VI.

II. RELATED WORK

The related work of the bus arbitration can be divided into three categories. First, implementing the existing bus arbitration protocol. Second, enhancing existing protocols in order to improve the whole bus-base system performance. Third, introducing new bus arbitration schemes. This section clarifies some of these works as follow:

Two new bus arbitration algorithms, which are Request-Service and Age-Based algorithms are introduced in [2]. The new algorithms try to improve the existing algorithms in term of latency caused by the contention among the bus masters. The Request-Service algorithm attempts to remove all forms of starvation among the competing masters. It also sets an upper limit for the waiting time for each master. The Age-Based algorithm gives more priority to masters that have recently used the bus, which will lead to improve the performance. The starvation problem is solved in this algorithm by using CritNo flag. Each algorithm has been implemented in a software simulation. The results show that the Request-Service model works well under low load. The Age-Based model performs well as the Futurebus model and reduces the amount of starvation and it is suitable when there is a need to transfer large blocks of data.

A HDL implementation and analysis of the lottery bus arbitration techniques are presented in [3]. The problem of generating a pseudo-random number greater than the total tickets value, which cause that none of the masters will get access the bus, is solved by allowing the bus to be granted to the master that is given highest priority. Moreover, the priority is rotated among the masters in order to prevent a single master to grant the bus for long time when the random number falls outside the range of the total tickets value. The results of the implementation indicate that dynamic lottery is more efficient than static lottery since it improves the average waiting time of

the bus masters. In addition, dynamic lottery using rotating priority ensures the best average waiting time for the bus masters comparing with other lottery approaches. However, more resources and on-chip power consumption are the most disadvantages of dynamic lottery comparing to static lottery.

A novel Dynamically Adaptive Arbitration (DAA) algorithm and compares it with the traditional bus arbitration protocols through using MPEG-4 video encoder application on FPGA instead of the analytical simulation methods are presented in [7]. The new DAA algorithm has been inspired by Lottery bus, where a dynamic algorithm has been implemented for centralized arbiter. The algorithm adaptively allocates the bus bandwidth to the masters that need it based on the usage history. The bus is offered more to those masters that have been the most active lately. The comparing results show that DAA competes with RR in performance sense in every evaluated case. DAA delivers the best performance when a high clock frequency is used. However, DAA drawback is the highest area requirement. If the area is an important issue, RR is a safe choice that performs well in most cases.

A dynamic round robin arbiter based on lottery method using VHDL is implemented in [8]. The results of the implemented model, which are shown on ModelSim tool, show that the latency is improved with the dynamic tickets more than the static tickets and the starvation is avoided. Moreover, the latency of the highest priority master is lower than that of some conventional architecture. The proposed arbiter provides flexible design for efficient SoC. However, the limitation of the dynamic method is that the distribution of random number is not uniform [9].

There are many other researches use FPGA and VHDL to implement and test their proposed or existing bus arbitration algorithms such as [10-13].

III. BUS ARBITRATION SCHEMES

The bus arbitration schemes can be divided into two broad categories, which are centralized and decentralized or distributed arbitration. In the centralized arbitration, there is a single arbiter for the bus. Each master sends its request to that arbiter, and then the arbiter decides which the bus owner according to the applied protocol is. In the decentralized arbitration, there is no explicit device or unit to decide which master will own the bus. However, all of the devices on the bus work together to determine which device will get the bus access [14]. The most knowing centralized bus arbitration protocols are daisy chain, static fixed priority, round robin, time division multiplexed, and lottery bus arbitration. In the following sub-sections, the round robin and lottery bus arbitration, which are related to the work in this paper, will be discussed.

A. Round Robin

A round robin (RR) protocol is a simple and fair arbitration style where no master is allowed to get the bus ownership indefinitely [15]. Any master wants to access the bus will get it in an arranged manner as shown in Fig. 1. Whenever a master's turn ends, either unused, because of the end of the data transfer, or limited time length, the turn is passed to the next master.

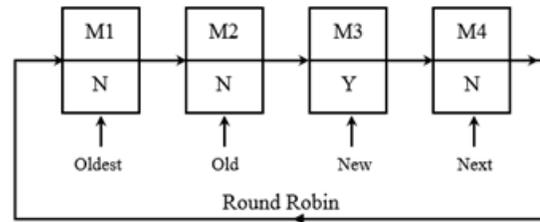


Fig. 1. Round Robin Arbitration

The RR has a disadvantage of checking all masters' interfaces even if they do not have pending requests. This action reduces the system performance as a result of bus distribution latency. Moreover, giving every master an equal share of the bus is not always a good idea. Because highly bus access masters will get scheduled as the idle masters [4, 7, 16].

The RR scheme can be improved by using a queue as shown in Fig. 2. This enhancement scheme has the same principle to serve all masters requests in an arranged manner. Instead of checking all masters' interfaces, it uses a queue to save the number of any master requests the shared bus. Then the masters' requests are served in First-In-First-Out (FIFO) manner. This scheme will be implemented in section VI under the name of queuing round robin (QRR) scheme.

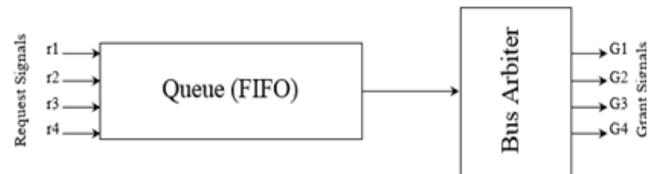


Fig. 2. Queuing Round Robin bus arbitration

B. Lottery Bus Arbitration

The role of the arbitration in the lottery bus arbitration algorithm is like a lottery manager that decides which lucky one can win the prize. The lottery manager accumulates the requests of the bus access from all of the masters. Each master is assigned a number of "lottery tickets". Then a pseudo random number is generated to choose one of the competing masters to be the winner of the lottery, favoring masters that have a larger number of tickets, and grant access is issued to the chosen master for a certain number of bus cycles. The random number guarantees that there is no master will monopolize the sharing resource [6, 9].

The inputs to the lottery manager are a set of requests and number of tickets held by each master. The output is a set of grant lines, one per master that defines which master had been allowed to access the bus.

According to the type of the tickets, lottery algorithms are divided into two types: static lottery and dynamic lottery [6]. In the static lottery, as shown in Fig. 3, each master has a fixed number of tickets. However, the number of tickets that is possessed by each master in the dynamic lottery are generated by a ticket generator, as shown in Fig. 4. For the both types, the same procedures are followed to decide the winner of the bus as the following:

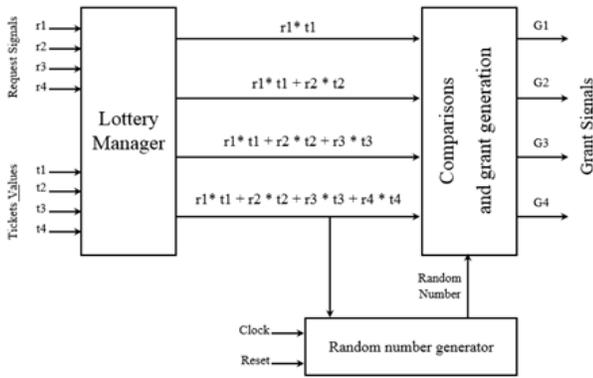


Fig. 3. Lottery arbiter with static tickets

- The lottery manager calculates the total tickets value for each master that has pending requests. This is given by $\sum_{j=1}^n r_j * t_j$, where n is the masters number, r is a Boolean variable represents the pending bus access request, and t is the number of tickets held by each master. For example, if the system has four processors and only three of them have pending requests, then n=4, r1=1, r2=0, r3=1, and r4=1. If the number of tickets that is possessed by each master are t1=1, t2=2, t3=3, and t4=4, then the total tickets values for processor1=1, processor2=1, processor3=4, and processor4=8.
- A pseudo-random number is generated in the range $[0, \sum_{j=1}^n r_j * t_j]$. It is supposed that the generated number is 5.
- If the generated number falls in the range $[0, r_1 * t_1]$, the bus is granted to master M1.
- In general, if the generated number lies in the range $[\sum_{k=1}^i r_k * t_k, \sum_{k=1}^{i+1} r_k * t_k]$ the bus is granted to master Mi+1. For our example, the generated number (5) falls in the range $[\sum_{k=1}^3 r_k * t_k, \sum_{k=1}^4 r_k * t_k] = [4, 8]$ so the bus is granted to processor4.

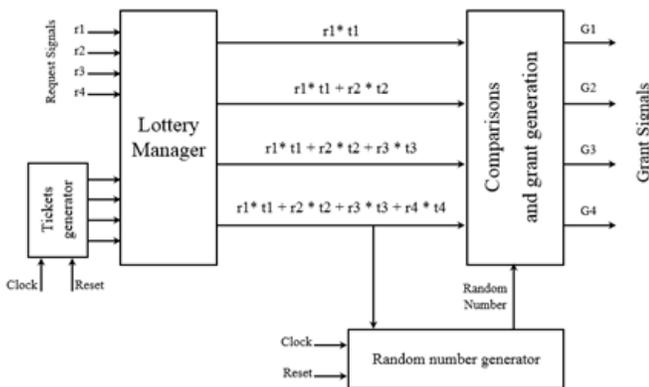


Fig. 4. Lottery arbiter with dynamic varying tickets

The advantages of the lottery algorithms are that all the masters that are requesting the bus get access to it (avoid starvation), and they improve the masters waiting time [3]. However, if the pseudo-random number is greater than the total

tickets value, none of the masters will get access the bus. Moreover, the fixed ticket values in the static lottery algorithm give high chance to masters with high ticket values [6]. The limitation of the dynamic lottery algorithm is that the distribution of the ticket values is non-uniform [9]. In addition, it is more complex and required extra logic to calculate the tickets of each master at run time [3].

IV. AGE-BASED LOTTERY ARBITRATION

As described in the related work section, the RR and the lottery bus arbitration compete in the performance sense. The developed scheme, in this work, represents the lottery bus arbitrations with additional enhancements to overcome their shortcoming.

The Age-Based Lottery (ABL), shown in Fig. 5, combines the dynamic lottery algorithm with the age-based algorithm from [2] to generate the ticket values. The ABL gives higher ticket values to masters that have recently won the bus. A preference during contention is given to the masters that are granted the bus recently. Each master has a ticket value can vary from 1 to MaxAge, which is a fixed parameter. The higher the ticket of a master, the more recently it has been granted the bus.

The algorithm shown in Fig. 6, illustrates the principle of ABL, which can be describe as follows: A CritNo flag is used for each master to balance the ticket value. When the master ticket value reaches MaxAge, the CritNo flag associated with that master is set. Then, if the ticket value is between the minimum age, which is 1, and MaxAge, then its ticket is decreased by one. If its ticket reaches 1, its flag is reset. If a master's flag is not set, its ticket is incremented by one after every bus grant. The integration between MaxAge and CritNo ensures a uniform distribution of the ticket values among the competing masters.

If there is only one request for the bus, the ABL will grant the bus to that request without any change on the corresponding ticket value since there is no any contention. On the other hand, the ticket values and CritNo flag must be changed when there are two masters or more compete on the bus. When there is more than one master request the bus and all of them reached the MaxAge, the associated ticket values reset to minimum age and their CritNo flags are reset.

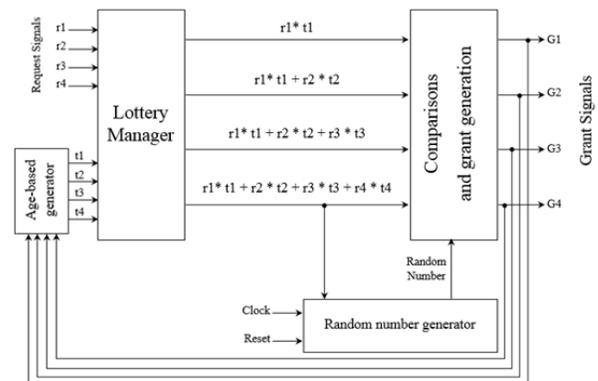


Fig. 5. Age-Based Lottery bus arbitration

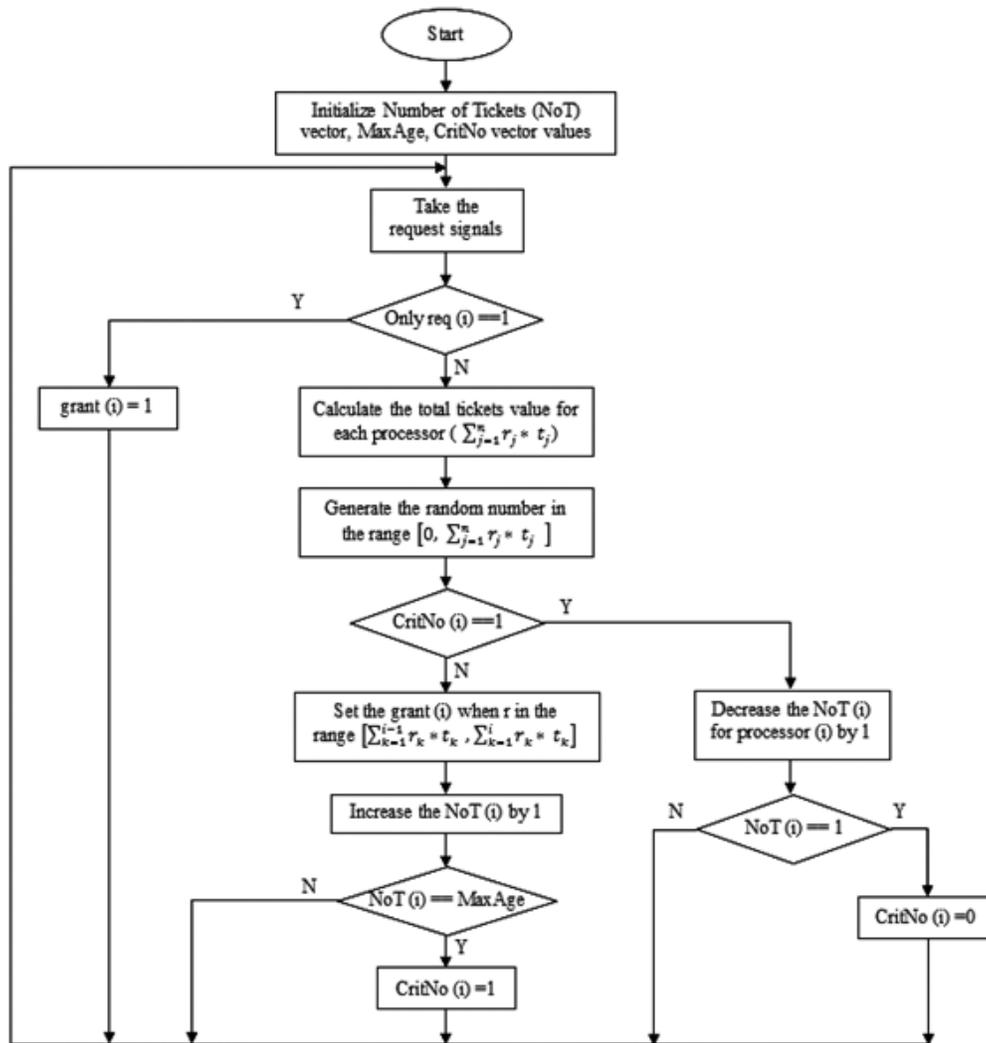


Fig. 6. The ABL algorithm for processor (i)

V. THE DEVELOPED BUS ARBITRATION SCHEME'S IMPLEMENTATION AND TESTING

To test the developed scheme and compare it to the traditional schemes, the following schemes are implemented for four processors (masters) using VHDL language:

- Traditional Round Robin (RR)
- Queuing Round Robin (QRR)
- Age-based lottery (ABL)
- Dynamic lottery (DL)
- Static lottery (SL)

The results of testing the developed scheme and the traditional schemes are obtained by a VHDL simulation tool from Mentor Graphics Company, which is called ModelSim. The ModelSim has an ability to illustrate the simulation results as a waveform, which is an easy way to recognize the required results. The main parameters of the comparison are the bus utilization, the bus distribution, and the latency.

To compare the tested bus arbitrations, the grant output signals are observed by providing input signals such as bus requests, clock, reset, and additional signals related to the arbitration type.

For more illustration, two testing scenarios of requesting the bus are applied on the tested bus arbitrations. First, when all the four masters request the shared bus. Second, when only two masters request that bus. The simulation runs 100,000 clock cycles. In every cycle, one processor takes the permission to access the bus.

The simulation results appear as shown in

TABLE I and TABLE II. For the bus utilization parameter, results show that all schemes utilize the shared bus effectively in the first testing scenario since all processors request the bus as shown in TABLE I. However, in the second testing scenario, the RR suffers from idle bus cycles that are given to processor number 2 and 3 as shown in TABLE II. These cycles affect the overall performance of the CMP. They must be granted to the requested processors to improve the overall performance. The rest schemes utilize the bus effectively in the second testing

scenario, too. They serve the requested processors only so there is no idle bus cycle.

TABLE I. THE FIRST TESTING RESULTS (ALL PROCESSOR REQUEST THE SHARED BUS)

Processor Arbitrer	1	2	3	4	Divergence
RR	25000	25000	25000	25000	0
QRR	25000	25000	25000	25000	0
ABL	21642	25509	29964	22885	3187.85
DL	30329	24616	25138	19917	3687.87
SL	15132	20069	29918	34881	7802.45

TABLE II. THE SECOND TESTING RESULTS (ONLY PROCESSOR 1 AND 4 REQUEST THE SHARED BUS)

Processor Arbitrer	1	2	3	4	Divergence
RR	25000	0	0	25000	0
QRR	50000	0	0	50000	0
ABL	49999	0	0	50001	1
DL	39181	0	0	60819	10819
SL	30172	0	0	69828	19828

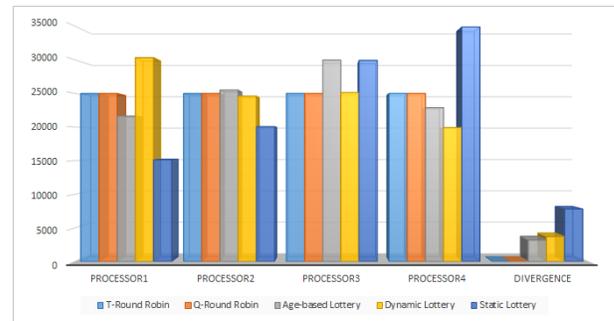
For the bus distribution parameter, results of the both testing scenarios show that the RR and QRR schemes surpass other schemes in the fair distribution. They give all processors the same priority degree to access the bus. In the first testing scenario, the ABL introduces fair distribution better than the dynamic and static lotteries. The ABL improves the distribution by 13.5% more than the DL and by 59% more than the SL. In the second testing scenario, the ABL has the same results of the QRR, which is better than the DL and SL by approximately 100%. Fig. 7 depicts the simulation results with the divergence in the bus distribution for each scheme.

For the latency parameter, the latency time is static in the RR and QRR schemes since each processor gets access to the shared bus in its order as shown in Fig. 7. However, there is no chance for any processor to get access to the shared bus for two or more cycles successively. This problem has been solved by using the lottery schemes. The latency time is improved using the probabilistic lottery schemes. Moreover, in ABL the latency time to get access to the shared bus is improved by the term of age as shown in Appendix A.

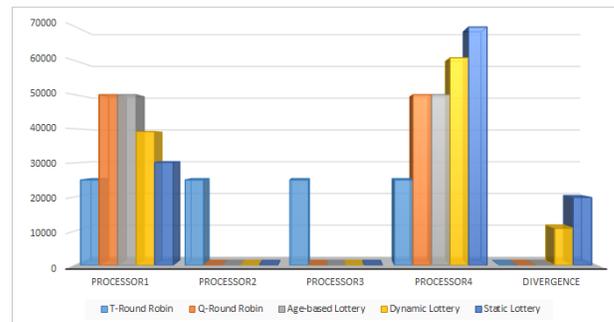
VI. CONCLUSION AND FUTURE WORK

In this paper, a new bus arbitration scheme, which is called an Age-Based Lottery (ABL), is developed to improve the shared bus utilization and distribution. The ABL is a new combination scheme that combines Lottery algorithm with Age-Based algorithm. The ABL is designed to overcome the traditional static lottery (SL) and dynamic lottery (DL) arbitrations shortcomings. The simulation results illustrate that the developed scheme improves the bus utilization and

distribution comparing to the traditional schemes by at least 13.5%.



a)The first testing results



b)The second testing results

Fig. 7. Simulation results with the divergence in the bus distribution for bus arbitration schemes

The shared bus in this paper limits the number of masters that can share it. This paper can be extended by designing an alternative bus implementation such as hierarchy of physical buses (tree bus) which may increase the number of masters in CMP systems.

REFERENCES

- [1] K. Olukotun, L. Hammond, and J. Laudon, "Chip multiprocessor architecture: techniques to improve throughput and latency," Synthesis Lectures on Computer Architecture, vol. 2, no. 1, pp. 1-145, 2007.
- [2] N. Ramasubramanian, P. Krishnan, and V. Kamakoti, "Studies on the Performance of Two New Bus Arbitration Schemes for MultiCore Processors," in International Advance Computing Conference (IACC), Patiala, pp. 1192-1196, 2009.
- [3] B. Tiwari, R. Chandraker, and N. Goel, "Comparative analysis of different lottery bus arbitration techniques for SoC communication," in 2016 International Conference on Computational Techniques in Information and Communication Technologies (ICCTICT), pp. 495-499, 2016.
- [4] J. Gupta and N. Goel, "Efficient Bus Arbitration Protocol for SoC Design," in International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM) India, pp. 396-400, 2015.
- [5] F. Wang, "Bus arbitration techniques to reduce access latency," ed: Google Patents, 2013.
- [6] P. Bajaj and D. Padole, "Arbitration schemes for multiprocessor Shared Bus," New Trends and Developments in Automotive Engineering, INTECH Publisher, 2011.
- [7] A. Kulmala, E. Salminen, and T. Hamalainen, "Distributed bus arbitration algorithm comparison on FPGA-based MPEG-4 multiprocessor system on chip," Computers & Digital Techniques, IET, vol. 2, no. 4, pp. 314-325, 2008.
- [8] D. Shanthi and R. Amutha, "Design of efficient on-chip communication architecture in MpSoC," in International Conference on Recent Trends

- in Information Technology (ICRTIT), Chennai, Tamil Nadu, pp. 364-369, 2011.
- [9] D. Shanthi and R. Amutha, "Design Approach to Implementation Of Arbitration Algorithm In Shared Bus Architectures (MPSoC)," Computer Engineering and Intelligent Systems, vol. 2, no. 4, pp. 185-196, 2011.
- [10] A. K. Singh, A. Shrivastava, and G. Tomar, "Design and Implementation of High Performance AHB Reconfigurable Arbiter for Onchip Bus Architecture," in International Conference on Communication Systems and Network Technologies (CSNT), Katra, Jammu, pp. 455-459, 2011.
- [11] D. Valle, G. Pablo, D. Atienza, I. Magan, J. G. Flores, E. A. Perez, et al., "Architectural exploration of MPSoC designs based on an FPGA emulation framework," in Proceedings of XXI Conference on Design of Circuits and Integrated Systems (DCIS), pp. 12-18, 2006.
- [12] W. Zhang, G.-M. Du, Y. Xu, M.-L. Gao, L.-F. Geng, B. Zhang, et al., "Design of a hierarchy-bus based MPSoC on FPGA," in 8th International Conference on Solid-State and Integrated Circuit Technology (ICSICT'06) Shanghai, pp. 1966-1968, 2006.
- [13] W.-T. Zhang, L.-F. Geng, D.-L. Zhang, G.-M. Du, M.-L. Gao, W. Zhang, et al., "Design of heterogeneous MPSoC on FPGA," in 7th International Conference on ASIC (ASICON'07) Guilin, pp. 102-105, 2007.
- [14] D. A. Patterson and J. L. Hennessy, Computer organization and design: the hardware/software interface, Newnes, 2013.
- [15] H. El-Rewini and M. Abd-El-Barr, Advanced computer architecture and parallel processing vol. 42, John Wiley & Sons, 2005.
- [16] I. S. Rajput and D. Gupta, "A Priority based Round Robin CPU Scheduling Algorithm for Real Time Systems," International Journal of Innovations in Engineering and Technology, vol. 1, no. 3, 2012.

APPENDIX A

SIMULATION WAVEFORM FOR THE AGE-BASED LOTTERY BUS ARBITRATION

